



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/975,113	10/11/2001	Clifford R. Johns	LUT 2 0059	8775
7590	06/07/2005		EXAMINER	
Richard J. Minnich, Esq. Fay, Sharpe, Fagan, Minnich & McKee, LLP. Seventh Floor 1100 Superior Avenue Cleveland, OH 44114			VU, TUAN A	
			ART UNIT	PAPER NUMBER
			2193	
			DATE MAILED: 06/07/2005	

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/975,113	JOHNS ET AL.
Examiner	Art Unit	
Tuan A. Vu	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 17 March 2005.
- 2a) This action is **FINAL**. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-15 is/are pending in the application.
 - 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-15 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)	5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)
Paper No(s)/Mail Date _____	6) <input type="checkbox"/> Other: _____

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 3/17/2005.

As indicated in Applicant's response, claim 10 has been amended. Claims 1-15 are pending in the office action.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1-15 are rejected under 35 U.S.C. 102(e) as being anticipated by Schaumont et al., USPN: 6,606,588 (hereinafter Schaumont).

As per claim 1, Schaumont discloses a method of designing hardware, comprising:
entering source code into source code file, said source code using a context-free grammar that describes a job the hardware being designed has to do (e.g. Vector model 307, Data-flow model 315 – Fig. 3; *behavioral description*, col. 3, lines 20-31; Figs. 6-10; col. 51, line 19 to col. 60, line 14; col. 65, lines 31-42; Fig. 21; *metacode generation* - col. 4, line 2 to col. 49, line 43 – Note: Vector modeling with rules firing upon input/output mapping and meta code describing SFG type graph based on input/token mapping according to a finite state machine read on context free grammar embodied as source file stored in Matlab model library – see col. 8, lines 2-4) rather than describing an implementation of the hardware; and

compiling the source code file to generate an output file which describes an optimized state machine for implementing the hardware (e.g. code 306 – Fig. 3; col. 75-226; *C++ code generation* – Fig. 22 – Note: simulation, scheduling, profiling and bench-marking amounts to generating *optimized* state machine responses – see col. 8, lines 32-44; cols. 17-36), said output file being written in C-based code.

As per claim 2, Schaumont discloses output file describing transmit and receive state machines (e.g. *sfg1*, *sfg2*, *sfg3*, *sfg fsm* - Fig. 7; FSM, component 1, FSM component 2 – Fig. 21 – Note: FSMs firing from rules after receiving some input from other FSM read on transmit and receive state machine associated with C++ code generated from descriptive source code).

As per claim 3, Schaumont discloses HDL and RTL (e.g. Descriptions 2206, 2208 -Fig. 22).

As per claim 4, Schaumont discloses, based on the SFG of source code: identifying sequences of states (e.g. col. 195, lines 293-310; col. 97 lines 63 to col. 105, line 390 – Note: collapsing state sequences, i.e. array element value, under one state dictated by a conditional or logical expression is equivalent to collapsing sequences under one state) in the source code occurring more than twice in a row (*loop_cycle[]* -col. 195 line 299); collapsing the sequence into one state (e.g *phi* <= *NCYC*, col. 195, line 306; *i=0*; *i<NF+F*, line 344 – Note: bounding all sequence of states being defined under one fsm controller control statement is equivalent to collapsing states into one condition whose state is determined from that that conditional statement); and wrapping a counter around that one state (*int phi* <= *NCYC*, col. 195, line 298; *i=0*; *i<NF*, line 335).

As per claim 5, Schaumont describes FSM and firing rules according to some data graph and vector, hence implicitly disclose grammar free context, hence inherently teaches Backus Naur Formalism.

As per claim 6, Schaumont discloses a computer-readable medium embodying a computer software for designing hardware, comprising:

a compiler which generates from source code an output file that describes an optimized state machine for implementing hardware (e.g. code 306 – Fig. 3; col. 75-226; *C++ code generation* – Fig. 22 – Note: simulation, scheduling, profiling and bench-marking amounts to generating optimized code – see cols. 17-36; col. 8, lines 32-44);

said source code using a context-free grammar that describes a job the hardware being designed has to do (e.g. Vector model 307, Data-flow model 315 – Fig. 3; *behavioral description*, col. 3, lines 20-31; Figs. 6-10; col. 51, line 19 to col. 60, line 14; col. 65, lines 31-42; Fig. 21; *metacode generation* - col. 4, line 2 to col. 49, line 43); and

said output file being written in a C-based language (*C++*, code 306 – Fig. 3).

As per claim 7, refer to rejection of claim 2.

As per claim 8, this claim corresponds to claim 4 above, and is rejected using the corresponding rejection as set forth therein.

As per claim 9, Schaumont discloses an user option that puts variables in the output file which have names matching those corresponding to source code node (e.g. col. 22, line 49 to col. 32, line 6), said variables getting set of defined value in every state generated from its corresponding node (e.g. col. 32, line 54 to col. 36, line 61 – Note: populating variables declared from the SFG nodes and providing for each variables being declared a value corresponding with

a state analysis from the SFG simulation process reads on matching variables corresponding to source code node and variables getting set of defined value in every state generated from SFG source model; col. 129, lines 230-267; col. 195, lines 323-350– Note: for each *maps[]* or *I_sample[]* or *Fq_coeff[]* element of a SFG a value is set)

As per claim 10, Schaumont discloses a system for designing hardware, said system comprising

computer means for entering an input file written in a source code using a context-free grammar which describes a job the hardware being designed has to do (e.g. Vector model 307, Data-flow model 315 – Fig. 3; *behavioral description*, col. 3, lines 20-31; Figs. 6-10; col. 51, line 19 to col. 60, line 14; col. 65, lines 31-42; Fig. 21; *metacode generation* - col. 4, line 2 to col. 49, line 43);

a computer compiler operative for selectively converting the input file into an output which is written in a C-based code, said output file describing an optimized state machine for implementing the hardware being designed (e.g. code 306 – Fig. 3; col. 75-226; *C++ code generation* – Fig. 22 – Note: simulation, scheduling, profiling and bench-marking amounts to generating optimized code – see cols. 17-36; col. 8, lines 32-44-- and scheduling in support of profiling implicitly disclose a selection of functions based on time efficiency observing rules or profile data).

As per claim 11, Schaumont discloses C-based code translated to HDL (e.g. Descriptions 2206, 2208 -Fig. 22).

As per claim 12, refer to claim 3 and col. 4, lines 25-37.

As per claim 13, Schaumont discloses selectively generating output files which describe both transmit and receive state machines (e.g. *sfg1*, *sfg2*, *sfg3*, *fsm 705-> sfg/fsm* → *fsm 712*-Fig. 7; FSM, component 1, FSM component 2 – Fig. 21 – Note: base on token and rules of the associated FSM and SFG for generating the C++ code is equivalent to selectively yielding results for implementing in code statements).

As per claim 14, this claim corresponds to claim 4 above, and is rejected using the corresponding rejection as set forth therein

As per claim 15, refer to claim 5.

Response to Arguments

4. Applicant's arguments filed 3/17/2005 have been fully considered but they are not persuasive. Following are Examiner's observations in regard thereto.

(A) Applicants have submitted that Schaumont begins with C++ description of system, hardware and architecture components as opposed to generating a C-based description of an optimized state machine, and that such C-based output is what Schaumont begins with instead of yielding as a result of compiling as from step 102 of Fig. 1 of the Application (Appl. Rmrks, pg. 5, bottom, pg. 6, top). The claim recites 2 limitations:

- (i) "entering source code into a source code file ... implementation of the hardware"; and
- (ii) "compiling the source code file to generate ... C-based code" ; leading to the following analysis.

As for (i), a source code using a context-free grammar and describing a job -- which the hardware is designed to do -- is being entered in a file. It is perceived from context free grammar being used by a source code that such source code uses some notations that help describe the

semantics of the source code language, reminiscent of the Backus-Naur Form representation used in parsing techniques for programming code. Further this source code (using this BNF type of representation for semantic analysis) describes a job some hardware is designed to do, which is interpreted as a description of functional requirements like in modeling. The source code used in this behavioral description (or functional requirement analysis) phase context amounts to a descriptive format (using some notational representation seen in tree parsing) which is broadly used for later implementation, thus a modeling specification file can read into that. The rejection has pointed to Schaumont's parts teaching a vector model file or a Matlab file, a model, a flow, a behavioral description (e.g. Vector model 307, Data-flow model 315, *architecture model* 316 – Fig. 3; *behavioral description*, col. 3, lines 20-31; col. 51). Because programming semantics can only be analyzed and parsed using a context free notation, the C/C++ constructs depicting such hardware behavior by Schaumont have met the requirement of the above limitation in view of one skill in the art broad and reasonable interpretation. There is no explicit requirement from the claim that obliges that said description (source file) of a hardware functional behavior has to be strictly in a particular language, e.g. Java, natural language, Handel C, C++, Pascal, Fortran, Matlab or in IDL; and so long as such description file is in source file using a context free form (equated to being parsed using some BNF notations, a commonly accepted form used in C compiler) like Schaumont's modeling language (e.g. based on MathLab libraries to describe hardware functional behavior) expressed via C/C++ constructs, such description file reads on the limitation.

As for (ii), the above source file is being compiled to generate a C-based output code describing optimized state machine for implementing the hardware. The rejection has pointed to

the generation of a C++ code as a result of compiling all the description files from limitation (i) above; and that reads on compiling of source code file. Because such compiled code comes from a C++ based system, there is no denial that this output is a C-based code. Further the rejection has shown why the state machine being part of the input for this compilation process encompasses all the optimization methodologies (see Schaumont: code 306 – Fig. 3; col. 75-226; *C++ code generation* – Fig. 22 – Note: simulation, scheduling, profiling and benchmarking amounts to generating *optimized* state machine responses – see col. 8, lines 32-44; cols. 17-36), which reads on optimized state machine. Hence, Schaumont has met limitation (ii). In all, based on the language of the claim, there is no clear requirement that limitations recited as (i) or (ii) be mapped exactly to what Applicants referred to in Fig. 1 of the Application (Application (Appl. Rmrks, pg. 6, top). For the sake of argument, if the claim lends the understanding that via compiling a C code output is generated; then nowhere in the fact that C++ source file is being compiled to yield a C++ output one skill in the art can find the sort of flaw that Applicants try to put forth because it would be quite plausible that to yield a C output code the input to the compiler be also a C source code, a language operating based on context free notation exactly like the C++ source code from Schaumont in Fig. 1D. One skill in the art interprets the claim in light of the broad and reasonable understanding pertinent an ordinary skill level in the art at the time the invention was made; and as such, there is no language specification requirement in (i) nor is there any lack by Schaumont on compiling to yield a C-based code as in (ii).

(B) Applicants have submitted that Schaumont does not teach claim 1 source code, claim 6 compiler, claim 10 input source file or compiler converting said source code (Appl. Rmrks, pg.

6, 2nd, 3rd, 4th para). The arguments fall under the scope of the issues being addressed above in section A; and are referred thereto.

(C) Applicants have submitted that the C++ OCAPI by Schaumont is not capable of generating both transmit and receive state machines from the same source code (Appl. Rmrks, pg. 6, bottom, pg. 7, top). The teaching shown in the scenario when a library file describing the behavior of the target system is inputted into Schaumont's compiling process to yield the flow graph and state machines as shown in Fig. 7 for instance reflects how claim 2 as recited is being construed and interpreted. The fsm elements shown in Fig. 7 exhibit receive fsm and a transmit fsm via the flow from block 705, path 711, and block 712; with fsm 705 transmitting and block sfg/fsm receiving or sfg/fsm firing and fsm 712 receiving. These fsm come from a same simulation and emulating information gathered from the source file then implemented in hardware construct code as mentioned in section A above. And the argument that Schaumont has to have two separate designs to implement a transmitter and receiver is not persuasive. The rejection has shown where Schaumont has met the concepts understood from the very teaching construed from the claim; and Applicants have failed to show why the cited parts are inappropriate.

The claims will stand rejected as set forth above.

Conclusion

5. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

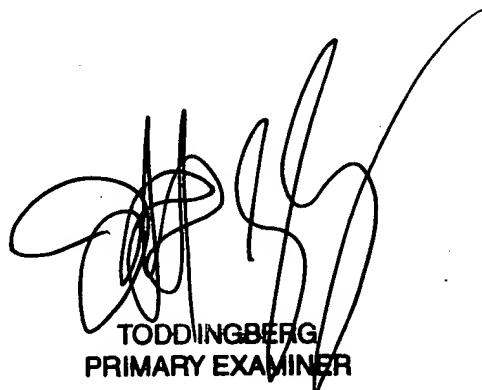
If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence – please consult Examiner before using) or 703-872-9306 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT
May 25, 2005



TODD NINGBERG
PRIMARY EXAMINER